

A Graph-Based Approach towards Discerning Inherent Structures in a Digital Library of Formal Mathematics

Lori Lorigo¹, Jon Kleinberg¹, Richard Eaton¹, Robert Constable¹

¹ Department of Computer Science, Cornell University, Ithaca, NY
{lolorigo, kleinber, eaton, rc}@cs.cornell.edu
<http://www.cs.cornell.edu>

2004

Abstract. As the amount of online formal mathematical content grows, for example through active efforts such as the Mathweb [21], MOWGLI [4], Formal Digital Library, or FDL [1], and others, it becomes increasingly valuable to find automated means to manage this data and capture semantics such as relatedness and significance. We apply graph-based approaches, such as HITS, or *Hyperlink-Induced Topic Search*, [11] used for World Wide Web document search and analysis, to formal mathematical data collections. The nodes of the graphs we analyze are theorems and definitions, and the links are logical dependencies. By exploiting this link structure, we show how one may extract organizational and relatedness information from a collection of digital formal math. We discuss the value of the information we can extract, yielding potential applications in math search tools, theorem proving, and education.

Keywords: mathematical knowledge management, formal methods, theorem provers, information seeking, information networks, link analysis, clustering

1 Introduction

Invaluable progress has been made in the development of digital libraries of mathematics [1, 3, 4, 15, 21]. Such progress also includes content and presentation-specific representations of mathematical knowledge [13, 14, 16], including architectures for exchanging mathematical content. These efforts are bringing together otherwise disparate collections of formal mathematics, and providing rich access to mathematical knowledge.

We are interested in providing services to users and designers of formal mathematical digital libraries. One such service is the ability to search for theorems or other mathematical objects with respect to their relationship to other theorems or objects in a given collection. The kinds of inter-object relationships may include similarity of theorem statements, similarity of proof methods, even similar levels of difficulty, particularly useful in user-model based approaches and education. Another service we wish to provide is finding core or basic theorems and axioms with respect to the

surrounding collection, theorems that cover or utilize heavily this core content, or theorems that seem to be *authoritative* and representative of a particular topic. Object dependencies of the kinds stored in the FDL can aid us in building these services.

Our approach is to use the mathematical objects and their logical dependencies to build a directed graph, and apply graph-theoretic algorithms to understand and extract information from the structure. Formal mathematical lemmas and their dependencies on other lemmas or definitions in their proofs form the nodes and edges of a directed graph respectively, as do web pages and hyperlinks on the Internet. Previous research has demonstrated effective methods for gathering relatedness and other semantic information about web pages on the Internet by operating on this directed graph. Popular eigenvector-based methods that are effective in web search by finding authoritative sources include Kleinberg’s HITS algorithm and Google’s PageRank [9]. In addition to abilities to rank important objects, capabilities to cluster or organize data into groups based on the graph structure have been developed and exploited. Web Trawling [12] uses a graph-theoretic approach to enumerate communities on the web, based on the findings of densely bipartite sub-graphs. Recent work in [17] finds communities in networks by iteratively removing detectable edges from the network to divide the collection into related groups.

Automatically categorizing or grouping related theorems in a formal digital library is one goal we are pursuing. In this work, however, we investigate, in general, the use of link analysis methods in the formal domain. While we restrict our studies to objects that were developed using the Nuprl5 refiner and are stored in the FDL, other interactive theorem provers make the same kinds of analysis possible, by having dependency information accessible. Such systems include Coq, MetaPRL, PVS, and others. Since we operate only on logical dependencies in this analysis, any collection of mathematics from which we can extract a dependency graph would be suitable.

We apply Kleinberg’s HITS algorithm to collections of formal mathematics in Section 2, where we also describe the link distributions of our chosen formal math data sets. In Section 3, we present a variation of HITS that has been tailored specifically for the formal math domain, observing varying levels of authoritativeness in this domain. We discuss future work in Section 4 and conclude in Section 5, including ways that this work opens possibilities for further automation and improvements in managing and understanding a digital library of mathematics.

2 HITS and Dependency Graphs

In the World Wide Web domain, *hubs* are web pages that point to a large number of authorities and *authorities* are pages pointed to by a large number of hubs. In the mathematical domain, our intuition tells us that hubs are proofs or lemmas that depend logically on a large number of authoritative lemmas or definitions, and authorities are the core definitions or theorems that are depended upon by a large number of hubs.

Kleinberg’s HITS algorithm demonstrates how to find hubs and authorities on the web. After a base set of web pages and links is generated, a directed graph, $G = (V, E)$, is constructed where the vertices, V , are the web pages, and the edges, E , are the hyperlinks. Hubs and authorities are found by assigning hub and authority weights to the vertices, and updating these values for k iterations, for some large enough k so that the process approaches equilibrium, or the weight vectors become nearly stable. If A is the adjacency matrix of the graph G , and vectors x and y are the authority and hub weight vectors, then x and y converge to the principle eigenvectors of $A^T A$ and $A A^T$ respectively. Then, the web pages with the c largest coordinates in x and in y when they have converged are deemed to be the c best authorities and hubs respectively.

2.1 Implementation and Design Choices

We implemented the algorithm in LISP, and ran the code inside of Cornell’s FDL on two different collections of formal mathematics that belong to that library: the Nuprl5 Standard collection [2], and the Event Structures collection [8]. These two collections were easily accessible to us and also presented a good contrast for measuring and evaluating our results. Large collections of PVS content also currently reside in the FDL, which would have been nice to contrast with Nuprl libraries, but we did not have the dependency information yet accessible for the PVS material, though in the future we hope to experiment with other collections.

In constructing the graphs, we chose theorems and definitions to be vertices of the graph. In further analyses, we plan to add rules as well. Tactics, and code objects were also potential candidates and may of interest for different kinds of information, such as relating proof styles. We restricted the edges to be representative of *logical* dependences. Again, other kinds of dependencies, including pointers to documents or comment objects, could be of interest as these links become more prominent.

We considered how to define a *logical dependency* in the context of this link analysis. We took the logical dependencies of a theorem to be all of the objects that were needed to complete the proof of the theorem. These include definitions and other theorems. It is a matter of style whether a user chooses to create a new theorem in order to prove a current one, or not. The former would create a dependency between two theorems where the latter would not, by self-containing the proof argument. We chose to use only the *direct* dependencies, as maintained by the FDL. By direct, we mean theorem A depends on theorem B if and only if B is directly pointed to in the proof of A . Indirect dependencies of A would then include the direct dependencies of B . While growing the graph by adding dependencies based on dependency transitivity might account for the variable proof designs, we would lose the structures that mimic the development of the theory, and likewise, the progressive level of expertise needed to understand a theory. We thus define the out-links of a node to point to its direct logical dependencies.

For the theorems, logical dependencies were gathered from the *primitive proofs* of the theorems which were created during refinement in the Nuprl proof system. Primitive proofs typically have too much detail to be desirable for reading, and many users prefer to read the tactic-level proofs. Nevertheless, primitive proofs contain *all* of the logical information of the proof execution. And in fact, the Nuprl primitive proofs are available online as part of the HELM project [3]. We do not include any out-links for the definitions. Definition objects do not have logical dependencies, that is, any definition is valid and they may only depend on a proof if for example, we would have considered extracts as definitions. However, definitions, like theorems are built up iteratively, and the definition for greatest common denominator, for example, depends on the definition of divides. These kinds of semantic dependencies are also accessible in the FDL. To understand authoritative objects with respect to definitions, we could capture the definition-definition dependencies, as these dependencies reveal when two definitions are related. The number of definitions with respect to theorems is small and we focus here on the logical links only.

2.2 Nuprl5 Standard Collection

The Nuprl5 Standard collection contains theories about integers, numbers, lists, booleans, and more. Information about its dependency graph follows in Table 1.

Table 1. Nuprl5 Standard Dependency Graph Data. Most fields should be clear. The *assortativity* of a network was defined by Newman in [18] and is a measure of the variance of the link distribution, or degree-degree correlations often used in social-network analysis. In practice, r is positive for social networks, where the nodes represent people, and negative for non-social networks such as the world wide web, power grids, and biological networks.

Nodes	Thms	Defs	Max. Out-links	Max. In-links	Edges	Assortativity
811	646	165	58	637	8765	-0.2949

Log-log density plots (using natural log) for the link distributions are shown below.

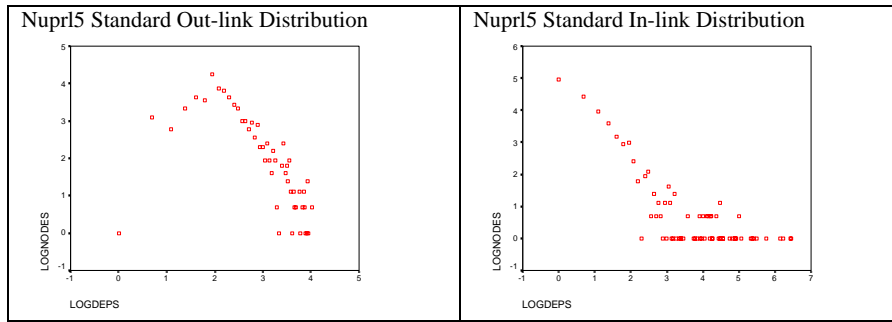


Fig. 1. Nuprl5 Standard Log-Log Density Plots.

We observe a short incline until a peak is reached and then a decline after a peak value is met in the out-link distribution. This demonstrates a characteristic number of

dependencies for the theorems. Though difficult to observe with such large variation, the tails are much longer for the in-links (we removed the end of the tail so that more data was visible since there we definition nodes with up to over 600 in-links), which happens to be typical of the World Wide Web links.

The power law distribution, which appears as a straight line in log-log density plots, is prevalent in many growing real-world networks, such as power grids, and the Internet [5]. Our data does not closely fit a power law over the entire set of degrees, but it is nearly linear over certain intervals of degree values: after the peak in the out-link graph, and also in the earlier part of the in-link graph.

Cumulative distributions shed more visual information as to the shape and nature of our graphs. Log-log plots of the cumulative distributions are shown in Figure 2 below.

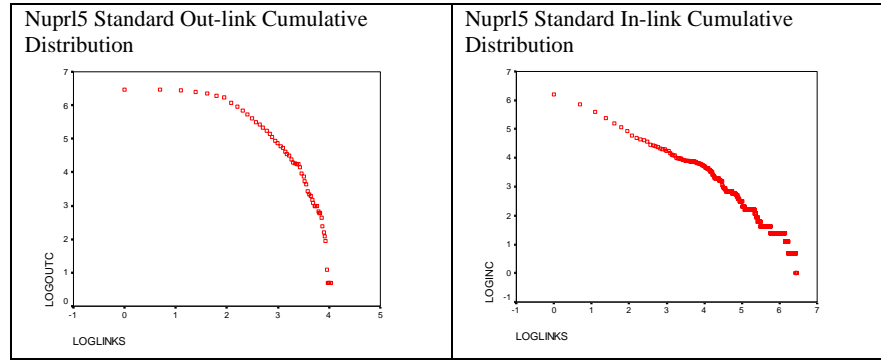


Fig. 2. Nuprl5 Standard Cumulative Log-Log Plots.

In the graph on the left, we observe a peak around 5-6, which is the same peak visible in the earlier graph. Also, the lower range of the in-link function follows a straight line, after which the data is very noisy, as shown in Figure 1 on the right.

From the above graphs, we observe a characteristic peak in the theorem out-links. Where this peak occurs may vary depending on the theory topic, or the complexity.

2.2 Nuprl 5 Standard HITS Results

In this section we include results from applying the HITS algorithm to our network.

The names of the top hubs and authorities of the Nuprl5 Standard collection of the FDL are listed below in decreasing order. As expected, the authorities are core, simple definitions, and the hubs are major theorems which depend on them.

Table 2. Nuprl5 Standard Hubs and Authorities

Authorities	member, all, prop, implies, and, iff, rev_implies, false,
Hubs	rem_mag_bound, select_listify_id, listify_wf, rem_eq_args_z, rem_base_case_z, select_firstn, listify_length, mod_bounds, modulus_wf

The authorities are the most fundamental and critical definitions in the Nuprl type theory. Hubs are objects from the list theory and the integer theory, two dominant theories in the collection. The authority values were primarily those with the greatest in-links, and likewise the hubs had large numbers of out-links. This was a trend, but not an exact representation. Rem_mag_bound, which states that the remainder of a divided by n is less than n , for example, has 51 out-links while the maximum has 56.

In addition to finding hubs and authorities, the HITS algorithm presents an eigenvector-based approach to finding clusters or communities in a graph. While the hubs and authorities can be found from principal eigenvectors, the best hubs and the best authorities from the non-principal eigenvectors of $A^T A$ and AA^T reveal communities. Furthermore, the communities that correspond to greater eigenvalues, are typically stronger. We looked for the vertices with the greatest hub weight and authority weight in the non-principal eigenvectors with the greater eigenvalues. These often are semantically related in practice. While the individual non-principal eigenvectors are thus a direct way to expose further structure in the data, we note that Ng et al. [19] observe instabilities in some cases in the use of these eigenvectors, and recommend more generally studying subspaces spanned by the non-principal eigenvectors.

Since the Nuprl Standard library has already been structured by humans around topics including lists, booleans, integers, and more, we compared the clusters found via the HITS method to the human-made categories, or topics. In some cases, the communities found could be considered as further refinements on the human-made topic structure. We set threshold values based on trial and error (an optimal threshold can perhaps be learned) as to when the corresponding eigenvalue was large enough to represent a community. The first four communities found are listed below.

Table 3. Nuprl5 Standard Communities

Community 1	listify_length, select_listify_id, int_seg_ind, select_append_front, decidable_ex_int_seg, or, decidable, so_apply1
Community 2	fincr_formation, fincr_wf, fincr_wf2, equiv_rel_functionality_wrt_iff
Community 3	fib_coprime, gcd_sat_gcd_p, gcd_sat_pred, fib_wf, gcd_wf, ycomb, not_wf
Community 4	atomic_char, assert_of_eq_int, prime_elim, assert_of_eq_atom, le_wf

The first is about lists, the second about recursive functions, and the latter two about Fibonacci numbers and atomicity in number theory. The objects in the clusters fell mainly in the human-made number theory category, and were often refinements on these human made categories.

2.2.1 Nuprl 5 Standard Number Theory Example

Hub and authority values can often best be utilized when you have already chosen a topic, as is the case in their use in Internet search. Typically, HITS seeds its search by growing its starting set of nodes. We extracted a sub-collection from Nuprl5 Standard about numbers, seeded this collection by recursively adding dependencies, and ran the HITS algorithm with the new subgraph. Information about this subgraph is in Table 4 below.

Table 4. Nuprl5 Number Theory

Nodes	Thms	Defs	Max. Out-links	Max. In-links	Edges	Assortativity
328	260	68	56	257	3397	-0.2848

The entire graph, dense due to many connections to core definitions is difficult to visualize in a confined space. Removing all of the definitions from the graph, as well as well-formedness theorems, which correspond to each definition to say it's well formed, we obtain a more visible graph in Figure 3.



Fig. 3. Number Theory Theorem Dependencies

The top hubs and authorities of the Nuprl5 Number theory collection including all of its dependencies are listed below in decreasing order.

Table 5. Nuprl5 Number Theory Hubs and Authorities.

Authorities	member, all, implies, prop, and, iff, rev_implies, false, not
Hubs	rem_mag_bound, gcd_sat_gcd_p, gcd_sat_pred, fib_coprime, fib_wf, absval_elim, absval_pos, absval_eq, gcd_ex_n

Since we seeded our search by adding objects that Number theory depends on, our authorities are similar to those from the entire Standard collection. If we wanted to know core definitions within number theory, we can eliminate the seeding step to get the following authorities: divides_wf, gcd_p, assoced, divides, assoced_weakening, assoced_wf.

Some of communities found from the HITS' eigenvector approach are listed below.

Table 6. Nuprl5 Number Theory Communities.

Community 1	divides_of_absvals, absval_assoced, absval_wf
Community 2	chrem_exists_aux_a, gcd_ex_n, chrem_exists_aux, atomic_char, prime_elim, gcd_exists_n, bezout_ident_n, chrem_exists
Community 3	div_3_to_1, div_2_to_1, div_4_to_1, divide_wf, nequal
Community 4	eqff_to_assert, eqtt_to_assert, assert_of_bnot, assert_of_band, prop

These communities are about absolute value, Chinese remainder theorem, division, and assertion respectively. Again, the contents are not so relevant. However, we observe that the names of objects within the groupings tend to be quite similar to one another, and these similarities in naming can be taken as indicative of human judgment that the objects themselves are related.

2.3 Event Structures Collection

This Event Structures collection was developed by Dr. Mark Bickford, a member of both ATC-NY, a subsidiary of Architecture Technology Corporation, and Cornell's PRL Group. The objects in it define and support a logic of events, describing a semantics for distributed systems. Some information about the dependency graph of this collection is included in Table 7 below.

Table 7. Event Structures Dependency Graph Data. Fields are the same as in Table 1.

Nodes	Thms	Defs	Max. Out-links	Max. In-links	Edges	Assortativity
1795	1306	489	210	708	16648	-0.15896

Log-log density plots (using natural log) for the out-links and in-links are shown in Figure 4 below.

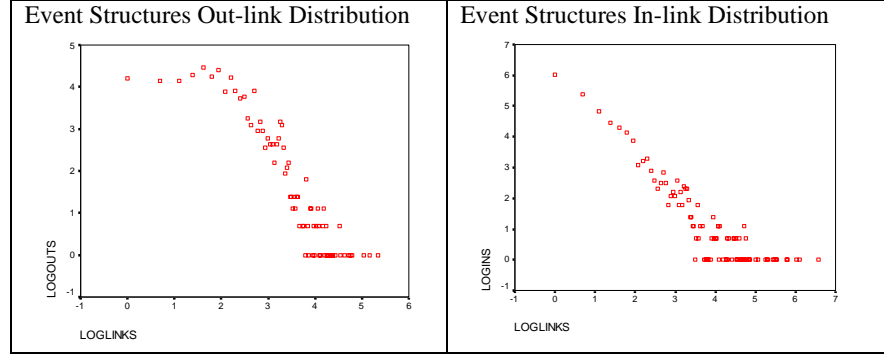


Fig. 4. Event Structures Log-Log Density Plots.

As in Figure 3, the functions plotted in Figure 4 are not quite linear, though probably are in a subset range. The data is noisy for objects above some link threshold. Again, we look at the log-log cumulative distributions for better clarity in Figure 5.

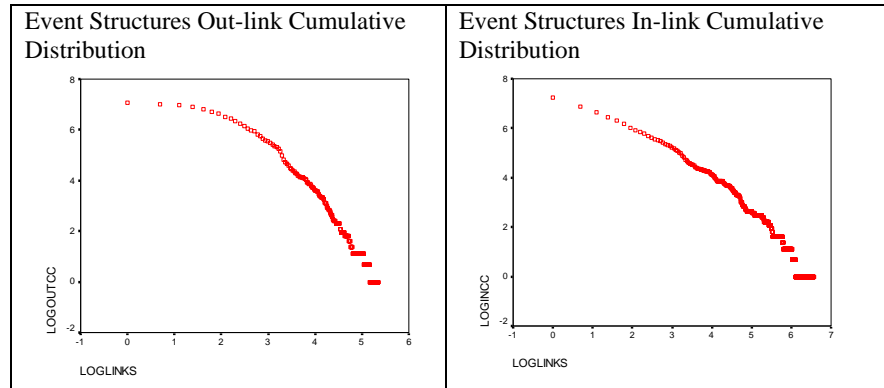


Fig. 5. Event Structures Cumulative Log-Log Plots.

The out-link distribution is steeper for the Event Structures collection than for the Nuprl Standard. Two factors could contribute to this. First, the Event Structures is actually built on top of the Standard collection, though the Standard collection is not included in its graph. It is an advanced collection, with large proofs about a specific topic, rather than a basic introductory collection that has more gradual variability. Second, the Nuprl Standard collection was built by several developers, and Dr. Bickford was the sole developer of the Event Structure proofs. He may have a proof signature style that may be somewhat evident in various structures of the graph. There is less of an initial incline in the out-link distribution, which is likely due to the fact that was built on top of another collection. The tails are again much longer for the in-links than for the out-links.

2.2 Event Structures HITS Results

Below we include results from applying Kleinberg’s HITS algorithm to our network.

The names of the top hubs and authorities of the Event Structures collection are listed below in decreasing order. The names of the objects are less informative than in the Nuprl5 Standard, but the objects’ contents are not entirely relevant. The authorities are fundamental lemmas, and the hubs are larger proofs which depend on them.

Table 6. Event Structures Hubs and Authorities.

Authorities	Id_wf, Knd_wf, IdLnk_wf, id-deq_wf, fpf_wf, fpf-cap_wf, Kind-deq_wf, fpf-dom_wf, fdf-trivial-subtype-top
Hubs	R-compat-base, R-Feasible-Dsys, sends-rule, pre-rule, d-feasible-world, R-sends-rule, R-interface-base, R-Feasible-action, R-Dsys-base-wf

Several communities found are listed below. It is difficult to measure the strength of these communities. We include only their names, inferring that the developer used a naming scheme such that similarly named objects are similar. While it is not apparent what the communities below entail, it is evident that they are related by name.

Table 6. Event Structures Communities.

Community 1	d-feasible-world, better-d-comp-step, d-comp-step2, d-comp-step, d-comp_wf, deq_wf
Community 2	Rpreinit-P_wf, Rpreinit-init_wf, Rpreinit-ds_wf, Rpreinit-T_wf, Rpreinit-loc_wf, Rpreinit-a_wf, Rpreinit?, Rpreinit?_wf, Rpreinit-ds, Rpreinit
Community 3	Reffect-f_wf, Reffect-ds_wf, Reffect-x_wf, Reffect-T_wf, Reffect-loc_wf, Reffect-knd_wf, Reffect?, Reffect?_wf
Community 4	Rframe-loc_wf, Rframe-T_wf, Rframe-L_wf, Rframe-x_wf, Rframe?, Rframe?_wf
Community 5	l_contains_disjoint, l_contains_append3, l_contains_append2, l_contains_append, l_contains_wf, l_contains-append4, l_contains

3 Stratified Authority Weighting

While the hubs and authorities method assigns weights to vertices at only two levels (hub and authority), we note that there are intermediate levels of authoritativeness in the formal math domain. Math collections, unlike collections of pages on the web, consist of a small number of known kinds of objects that demonstrate varied discrete levels of authoritativeness. These kinds, or categories, are definitions, rules, theorems, proofs, and perhaps others, depending on the theorem prover and its implementation, such as extracts or hypotheses, or conjectures. Furthermore, formal theories are often built up in a step-like, or modular fashion. For these reasons, we wish to consider *levels* of authority.

3.1 Authority and Hub Weight Levels

We observe that the authority and hub weight values don't necessarily degrade naturally, and instead the graphs look somewhat step-like. Figure 6 shows a plot of the authority weights for the Nuprl5 Standard library definitions and theorems.

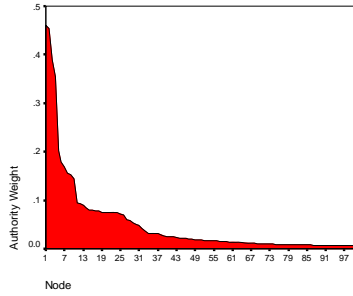


Fig. 6. Authority Weights

There are a couple small minor plateaus. The lower plateau, at around .7 shows the importance of the booleans, including objects such as `bfalse`, `btrue`, `or`, `ifthenelse`, and `true_wf`.

Next we consider only theorems, excluding wellformedness theorem, which yields step-like hub weighting. Figure 7 shows hub weights for a subset of standard theorems including integer theories, number theory, list theory, and their dependencies.

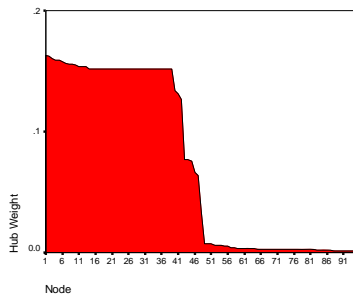


Fig. 7. Hub Weights

Since the Standard library is a basic collection, we would expect to find more levels when combining collections built on top of Standard. Dr. Bickford has built an additional list theory, on top of the Standard library and we plot the hub and authority weights for this collection, combined with its dependencies in Figure 8.

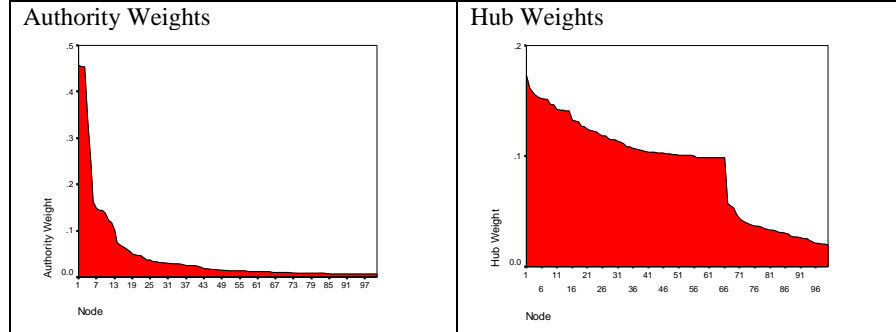


Fig. 8. Extended List Theory Hub and Authority Weights

We do observe slightly more structure in the hubs graph for list theory, showing its modularity. In comparing these graphs to Figures 6 and 7, we observe that there is a potentially characteristic difference between the distribution of hub weights and the distribution of authority weights.

3.2 Dependency-Based Levels

Instead of using HITS to find levels, we can a priori define levels based on the dependencies of the objects. We adopt the following definition of level.

Level 0 contains all objects that have no dependencies.
 Level i contains all objects, x , such that x is not in level j for any $j < i$, and x depends only on objects in levels $0 \dots k$ for $k < j$.

Using this definition of levels, the Nuprl5 Standard collection contained a total of 19 levels. All of the definitions were in Level 0. In the Event Structures library there were a total of 31 levels. Distributions of the levels are shown in the Figure 9 below.

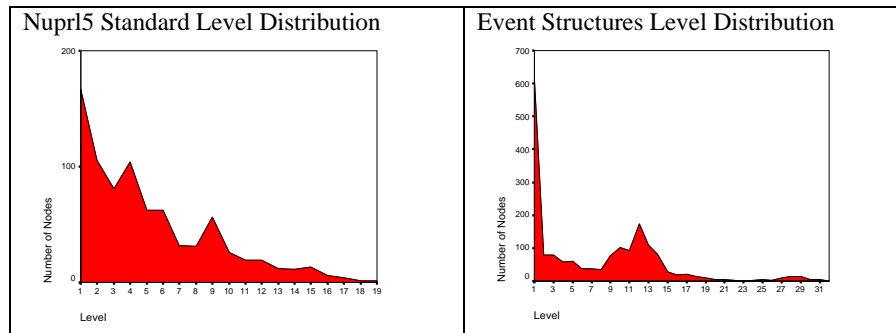


Fig. 9. Level Distribution

The high peaks are most interesting. These graphs reiterate in depth what we saw earlier in breadth that there is some characteristic depth (level) of proofs, just as there

was in breadth (link degree). One's proof style may influence the level where there is a peak. The peak on the right is particularly noticeable; it suggests that when stratified by depth, the graph exhibits a "wide" region in the middle. In this way, it can potentially be viewed as an interesting analogue, for acyclic graphs, of the "bow-tie" model of the Web [10].

In order to extract information from these levels, one may iteratively run the HITS algorithm, iteratively finding the hubs and authorities for the graph containing the union of level i and $i+1$, for $i = 0 \dots m-1$, where m is the maximum number of levels. The internal HITS algorithm which runs on the adjacency matrix of a graph remains unchanged, but rather iterations of it are made according to external dependency data. The authors leave this for future work, speculating that the hubs of level i and the authorities of level $i+1$ may be similar.

5 Ongoing and Future Work

This work has opened for us a number of questions with applications to digital libraries of formal mathematics. We hope to be able to answer deeper questions than the hubs, authorities, and communities discussed here, and to put some of our preliminary findings to practice and testing. We wish to extend these techniques to work on other representations of mathematical objects, and contribute towards ongoing work in formal mathematical representation and presentation. Perhaps the strongest benefits may come from combining this graph-based approach with others that reveal different kinds of information when it is available, such as integrating authority measures with pattern-matching based search when searching a library of math.

Additionally, this work shows promise for an automated way of merging two collections of math from different proof assistants, by first matching up core basic definitions and then using dependency-graph analysis information to find similar proofs from two separate proof assistants. We may also use it to organize a single library, or to prepare a library for presentation by asking which objects are most influential and ought to be documented? Also, tools to visualize and interact with the digital collections surely aid us in analyzing the structures. Several theorem provers offer tree-like user interfaces and work in [7] presents methods for pruning the dependency graphs so that they are visually appealing. We are interested in how visualization in this context can aid a user searching a digital library of mathematics.

Furthermore, the scale of the amount of mathematical data is surely not even comparable to the size of the web. The HITS methods are tractable for very large data sets. We may be able to take advantage of our relatively small graph sizes to get improved results.

6 Conclusion

We have shown applications of WWW search techniques to a new domain, namely, formal methods, along with presenting a modified stratified method for extracting information specific for formal math. This work is exploratory and the examples above attempt to demonstrate the potential of applying the HITS algorithm and related approaches to categorizing and searching formal mathematics.

We presented two applications of HITS (1) finding Hubs and Authorities and (2) Communities, and also showed characteristics specific to the depth of proofs in the fdl library. We can conclude from these experiments that automated means can be used to discern hub, authority, and community structure in at least one particular library of formal mathematics. The automated extraction of these kinds of relationships can be useful in ranking search results, for example, in the case of the authorities, or in organizing or merging related theorems, for example, based on the community structure. These experiments also demonstrate that logical dependencies alone can be informative in extracting these relationships. Also, what was less expected were findings of strongly characteristic breadth (link-degree) and depth (level) of the library. We cannot conclude whether these characteristics are specific to Nuprl, or common to the formal math domain, but we provide here a good basis for future comparison and discovery.

Acknowledgments

The authors wish to thank Dr. Mark Bickford for making his libraries of formal mathematics accessible to us and open for analysis. The authors also thank Dr. Stuart Allen for valuable discussions and input in the areas of formal digital libraries and mathematical knowledge management. This work was supported in part by ONR Grant #N00014-01-1-0765 and National Science Foundation #333526.

References

1. S. Allen, M. Bickford, R. Constable, R. Eaton, C. Kreitz, L. Lorigo. FDL: A Prototype Formal Digital Library, Cornell University, Unpublished manuscript, 2002.
2. S. Allen, R. Constable, R. Eaton, C. Kreitz, and L. Lorigo. The Nuprl Open Logical Environment, In *Proceedings of 17th International Conference on Automated Deduction*, LNAI 1831, pp. 170-176, Springer-Verlag, 2000.
3. A. Asperti, L. Padovani, C. Sacerdoti Coen, F. Guidi, I. Schena. Mathematical Knowledge Management in HELM. *Annals of Mathematics and Artificial Intelligence, Special Issue on Mathematical Knowledge Management*, Vol. 38, Issue 1-3, pp. 27-46, ISSN 1012-2443, Kluwer Academic Publishers, 2003.
4. A. Asperti, B. Wegner. MOWGLI: A New Approach for the Content Description in Digital Documents. *Ninth International Conference "Crimea"*

- 2002" *"Libraries and Associations in the Transient World: New Technologies and New Forms of Cooperation"* 2002.
5. A. Barabasi. *"Linked: How Everything Is Connected to Everything Else and What It Means"*, Plume Books, 2003.
6. V. Batagelj, A. Mrvar. PAJEK, Program for Large Network Analysis, <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>.
7. Y. Bertot, O. Pons. Dependency graphs in Interactive Theorem Provers *INRIA Tech Report*, 2000.
8. M. Bickford, R. Constable. A Logic of Events, *Cornell University Technical Report 2003-1893*, 2003.
9. S. Brin, L. Page. The anatomy of a large-scale hypertextual (Web) search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.
10. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener. Graph structure in the web. In *Proceedings of the Ninth International World Wide Web Conference*, 2000.
11. J. Kleinberg. Authoritative Sources in a Hyperlinked Environment, In *Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms*, 1998.
12. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins. Trawling the Web for cyber communities, *Computer Networks*, 31: 1481-1493, 1999.
13. M. Kohlase. OMDoc: Towards an Internet Standard for the Administration, Distribution and Teaching of mathematical Knowledge. In *Proceedings of Artificial Intelligence and Symbolic Computation*, Springer LNAI, 2000.
14. M. Kohlase. OMDoc: An Infrastructure for OpenMath Content Dictionary Information. *Bulletin of the ACM Special Interest Group for Algorithmic Mathematics*, SIGSAM, 2000.
15. D. Lozier. The DLMF Project: A New Initiative in Classical Special Functions. *Proc. International Workshop on Special Functions - Asymptotics, Harmonic Analysis and Mathematical Physics*, 1999.
16. Mathematical Markup Language (MathML) 2.0, W3C Recommendation, 21 February 2001. <http://www.w3.org/TR/MathML2/>.
17. M. Newman, M. Girvan. Finding and evaluating community structure in networks, *Physical Review E* 69, 026113, 2004.
18. M. Newman, J. Park. Why social networks are different from other types of networks, *Physical Review E* 68, 036122, 2003.
19. Andrew Y. Ng, Alice X. Zheng, Michael I. Jordan. Stable Algorithms for Link Analysis. In *Proc. 24th Annual Intl. ACM SIGIR Conference*, 2001.
20. PRL Group, Department of Computer Science, Cornell University, <http://www.nuprl.org/Nuprl4.2/Libraries/Welcome.html>.
21. J. Zimmer, M. Kohlhase. System Description: The MathWeb Software Bus for Distributed Mathematical Reasoning. *Proceedings of the 18th Int'l Conference on Automated Deduction (CADE 18)*, LNAI 2392, Springer Verlag, 2002.